Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

MOTOROLA

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Embedded SDK (Software Development Kit)

MFCR2 Detection Library

SDK151/D
Rev. 0, 10/24/2002

© Motorola, Inc., 2002. All rights reserved.

MOTOROLA
intelligence everywhere

digital dna

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Contents

About This Document

Audience ix
Organization ix
Suggested Reading ix
Definitions, Acronyms, and Abbreviations xi
References xi

Chapter 1
Introduction

1.1 Quick Start 1-1
1.2 Overview of MFCR2 Detection 1-1
1.2.1 Background 1-1
1.2.2 Features and Performance 1-2

Chapter 2
Directory Structure

2.1 Required Core Directories 2-1
2.2 Optional (Domain-Specific) Directories 2-2

Chapter 3
MFCR2 Detection Library Interfaces

3.1 MFCR2 Detection Services 3-1
3.2 Interface 3-1
3.3 Specifications 3-4
3.3.1 MFCR2DetectCreate 3-5
3.3.2 MFCR2DetectInit 3-8
3.3.3 MFCR2Detect 3-10
3.3.4 MFCR2DetectDestroy 3-14

Chapter 4
Building the MFCR2 Detection Library

4.1 Building the MFCR2 Detection Library 4-1
4.1.1 Dependency Build 4-1
4.1.2 Direct Build 4-2

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 5
Linking Applications with the MFCR2 Detection Library

5.1 MFCR2 Detection Library 5-1
5.1.1 Library Sections 5-1

Chapter 6
MFCR2 Detection Applications

6.1 Test and Demo Applications. 6-1

Chapter 7
License

7.1 Limited Use License Agreement 7-1

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

List of Tables

Table 3-1 MFCR2DetectCreate Arguments 3-5
Table 3-2 MFCR2DetectInit Arguments. 3-8
Table 3-3 ,MFCR2Detect Arguments 3-10
Table 3-4 MFCR2DetectDestroy Arguments 3-14

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

List of Figures

Figure 2-1 Core Directories 2-1
Figure 2-2 DSP568XX Directories. 2-2
Figure 2-3 Mfcr2 Directory Structure 2-3
Figure 4-1 Dependency Build for MFCR2 Detection Library 4-1
Figure 4-2 MFCR2 Detection Project 4-2
Figure 4-3 Execute Make 4-2

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

List of Examples

Code Example 3-1 C Header File MFCR2Detect.h 3-1
Code Example 3-2 mem Library 3-5
Code Example 3-3 Use of MFCR2DetectCreate Interface 3-7
Code Example 3-4 Use of MFCR2DetectInit Interface. 3-8
Code Example 3-5 Use of MFCR2Detect Interface 3-12
Code Example 3-6 Use of MFCR2DetectDestroy Interface 3-14
Code Example 5-1 linker.cmd File 5-1

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

About This Document

This manual describes the Multi-Frequency Compelled R2 (MFCR2) Detection algorithm for use with Motorola's Embedded Software Development Kit, (SDK).

Audience

This document targets software developers implementing MFCR2 Detection within software applications.

Organization

This manual is arranged in the following sections:

- **Chapter 1, Introduction** —provides a brief overview of this document
- **Chapter 2, Directory Structure** —provides a description of the required core directories
- **Chapter 3, MFCR2 Detection Library Interfaces** —describes all of the MFCR2 Detection Library functions
- **Chapter 4, Building the MFCR2 Detection Library** —tells how to execute the system library project build
- **Chapter 5, Linking Applications with the MFCR2 Detection Library** —describes organization of the MFCR2 Detection Library
- **Chapter 6, MFCR2 Detection Applications** —describes the use of MFCR2 Detection Library through test applications
- **Chapter 7, License** —provides the license required to use this product

Suggested Reading

We recommend that you have a copy of the following references:

- *Motorola DSP56800 Family Manual,* DSP56800FM/AD
- *Motorola DSP56800E Reference Manual,* DSP56800ERM/D
- *Motorola DSP568xx User's Manual*, for the DSP device you're implementing
- *Inside CodeWarrior: Core Tools*, Metrowerks Corp.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Conventions

This document uses the following notational conventions:

Typeface, Symbol or Term	Meaning	Examples
`Courier Monospaced Type`	Code examples	`//Process command for line flash`
Italic	Directory names, project names, calls, functions, statements, procedures, routines, arguments, file names, applications, variables, directives, code snippets APIs in text	...and contains these core directories: *applications* contains applications software... ...CodeWarrior project, *3des.mcp* is... ...the *pConfig* argument.... ...defined in the C header file, *aec.h*....
Bold	Reference sources, paths, emphasis	...refer to the **Targeting DSP56F80x Platform** manual.... ...see: **C:\Program Files\Motorola\Embedded SDK\help\tutorials**
Blue Text	Linkable on-line	...refer to Chapter 7, License....
Number	Any number is considered a positive value, unless preceded by a minus symbol to signify a negative value	3V -10 DES^{-1}
ALL CAPITAL LETTERS	# defines/ defined constants	# define INCLUDE_STACK_CHECK
Brackets [...]	Function keys	...by pressing function key [F7]
Quotation marks, “...”	Returned messages	...the message, “Test Passed” is displayed.... ...if unsuccessful for any reason, it will return “NULL”...

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Definitions, Acronyms, and Abbreviations

The following list defines the acronyms and abbreviations used in this document. As this template develops, this list will be generated from the document. As we develop more group resources, these acronyms will be easily defined from a common acronym dictionary. Please note that while the acronyms are in solid caps, terms in the definition should be initial capped ONLY IF they are trademarked names or proper nouns.

DSP	Digital Signal Processor or Digital Signal Processing
I/O	Input/Output
IDE	Integrated Development Environment
MFCR2	Multi-Frequency Compelled Register Signaling
MIPS	Million Instructions Per Second
OnCE™	On-Chip Emulation
SDK	Software Development Kit
SRC	Source

References

The following sources were used to produce this book:

1. *Motorola DSP56800 Family Manual,* DSP56800FM/AD
2. *Motorola DSP56800E Reference Manual*, DSP56800ERM/D
3. *Motorola DSP568xx User's Manual*, for the DSP device you're implementing
4. *Targeting Motorola DSP568xx Platform*, for the DSP device you're implementing
5. *Motorola Embedded SDK Programmer's Guide,* SDK101/D

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 1
Introduction

Welcome to Motorola's Family of Digital Signal Processors (DSPs). This document describes the Multi-Frequency Compelled R2 (MFCR2) Detection Library, which is a part of Motorola's comprehensive Software Development Kit (SDK) for its DSPs. In this document, you will find all the information required to use and maintain the MFCR2 Detection Library interface and algorithms.

Motorola provides these algorithms to you for use with Motorola DSPs to expedite your application development and reduce the time it takes to bring your own products to market.

Motorola's MFCR2 Detection Library is licensed for your use on Motorola processors. Please refer to the standard Software License Agreement in **Chapter 7** for license terms and conditions; please consult with your Motorola representative for premium product licensing.

1.1 Quick Start

Motorola Embedded SDK is targeted to a large variety of hardware platforms. To take full advantage of a particular hardware platform, use **Quick Start** from the **Targeting DSP568xx Platform** documentation.

For example, the **Targeting DSP56826 Platform** manual provides more specific information and examples about this hardware architecture. If you are developing an application for the DSP56826EVM board or any other DSP56826 development system, refer to the **Targeting DSP56826 Platform** manual for **Quick Start** or other DSP56826-specific information.

1.2 Overview of MFCR2 Detection

During the exchange of multi-frequency combinations the link or multi-link section is terminated at both ends by equipment allowing multi-frequency combinations to be sent and received under mutual, compelled control, as indicated in Recommendation Q.440 of ITU-T. The MFCR2 library implements the receiving function of the multi-frequency signaling equipment.

1.2.1 Background

The composition of the multi-frequency combinations is specified in Recommendation Q.441 of ITU-T.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

The frequencies in the forward direction are:

- f_0 = 1380Hz
- f_1 = 1500Hz
- f_2 = 1620Hz
- f_3 = 1740Hz
- f_4 = 1860Hz
- f_5 = 1980Hz

The frequencies in the backward direction are:

- f_0 = 1140Hz
- f_1 = 1020Hz
- f_2 = 900Hz
- f_3 = 780Hz
- f_4 = 660Hz
- f_5 = 540Hz

1.2.2 Features and Performance

MFCR2 Detection is multi-channel and re-entrant.

For details on memory and MIPS for a particular DSP, refer to the **Libraries** chapter of the appropriate Targeting manual.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 2
Directory Structure

2.1 Required Core Directories

Figure 2-1 details required platform directories:

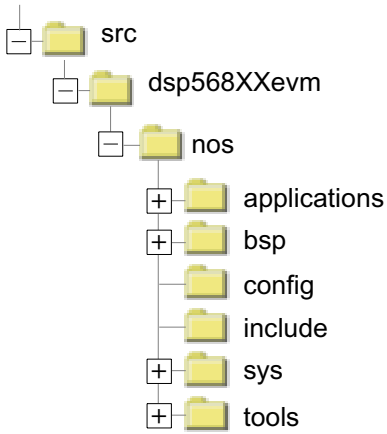

Figure 2-1. Core Directories

In **Figure 2-1**, DSP568XXEVM has no operating system (nos) support and contains these core directories:

- ***applications*** contains applications software that can be exercised on this platform
- ***bsp*** contains board support package specific for this platform
- ***config*** contains default HW/SW configuration for this platform
- ***include*** contains SDK header files which define the Application Programming Interface
- ***sys*** contains required system components
- ***tools*** contains utilities used by system components

There are also optional directories that include domain-specific libraries.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

2.2 Optional (Domain-Specific) Directories

Figure 2-2 demonstrates how the MFCR2 Detection algorithm, *Mfcr2*, is encapsulated in the domain-specific directory *telephony*.

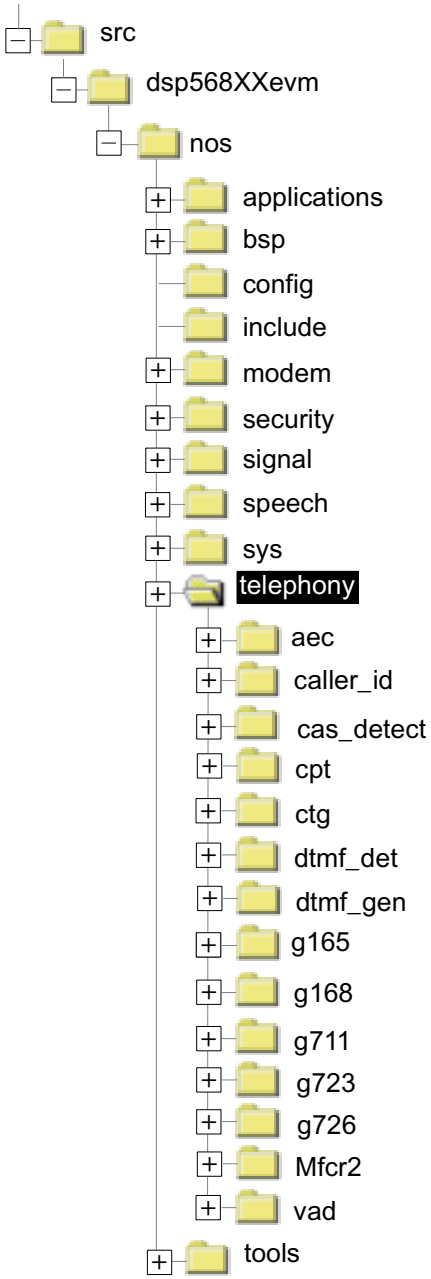

Figure 2-2. DSP568XX Directories

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

The *Mfcr2* directory includes MFCR2 Detection-specific algorithms. **Figure 2-3** shows the *Mfcr2* directory structure.

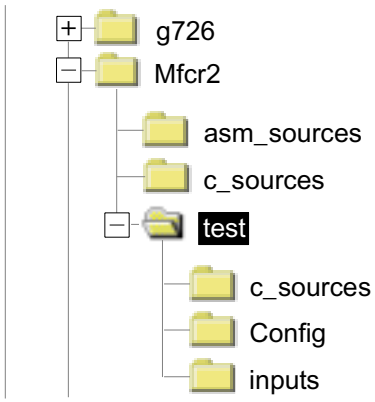

***Figure 2-3. Mfcr2* Directory Structure**

The *Mfcr2* directory includes these sub-directories:

- ***asm_sources*** includes all assembly source files
- ***c_sources*** includes C source files providing a C-callable API for MFCR2 Detection
- ***test*** includes C source files and configuration necessary for testing MFCR2 Detection library modules
  - — ***c_sources*** contains test code used to confirm correct operation of the MFCR2 Detection library
  - — ***Config*** contains configuration files *appconfig.c*, *appconfig.h* and *linker.cmd* specific to MFCR2 Detection testing
  - — ***inputs*** contains the test input files

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 3
MFCR2 Detection Library Interfaces

3.1 MFCR2 Detection Services

The MFCR2 signal consists of two frequencies taken either from the set of six forward frequencies or from the set of six backward frequencies. The MFCR2 receiver must detect any valid MFCR2 tone and output the numerical value corresponding to that tone.

3.2 Interface

The C interface for the MFCR2 Detection library service is defined in the C header file *MFCR2Detect.h,* shown in **Code Example 3-1**.

Code Example 3-1. C Header File *MFCR2Detect.h*

```
/* File: MFCR2Detect.h */

#ifndef __MFCR2DETECT_H
#define __MFCR2DETECT_H

/**************************
 Foundational Include Files
**************************/

#include "port.h"

/************************************
 #defines to be used to configure MFCR2
 ************************************/

/* These #defines should be used by the user to initialize the configuration
structure.*/

#define MFCR2_FORWARD                    0x0000
#define MFCR2_BACKWARD                   0x0001
```


For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
/**************************
 Errors returned by MFCR2
 *************************/
#define MFCR2_INVALID                       0x8000

/**************************
 Status returned by MFCR2
 *************************/

#define MFCR2_SIGNAL1_DETECTED              0x0001
#define MFCR2_SIGNAL2_DETECTED              0x0002
#define MFCR2_SIGNAL3_DETECTED              0x0003
#define MFCR2_SIGNAL4_DETECTED              0x0004
#define MFCR2_SIGNAL5_DETECTED              0x0005
#define MFCR2_SIGNAL6_DETECTED              0x0006
#define MFCR2_SIGNAL7_DETECTED              0x0007
#define MFCR2_SIGNAL8_DETECTED              0x0008
#define MFCR2_SIGNAL9_DETECTED              0x0009
#define MFCR2_SIGNAL10_DETECTED             0x000a
#define MFCR2_SIGNAL11_DETECTED             0x000b
#define MFCR2_SIGNAL12_DETECTED             0x000c
#define MFCR2_SIGNAL13_DETECTED             0x000d
#define MFCR2_SIGNAL14_DETECTED             0x000e
#define MFCR2_SIGNAL15_DETECTED             0x000f

#define MFCR2_STOP_DETECTED_WITH_SIGNAL1    0x0031
#define MFCR2_STOP_DETECTED_WITH_SIGNAL2    0x0032
#define MFCR2_STOP_DETECTED_WITH_SIGNAL3    0x0033
#define MFCR2_STOP_DETECTED_WITH_SIGNAL4    0x0034
#define MFCR2_STOP_DETECTED_WITH_SIGNAL5    0x0035
#define MFCR2_STOP_DETECTED_WITH_SIGNAL6    0x0036
#define MFCR2_STOP_DETECTED_WITH_SIGNAL7    0x0037
#define MFCR2_STOP_DETECTED_WITH_SIGNAL8    0x0038
#define MFCR2_STOP_DETECTED_WITH_SIGNAL9    0x0039
#define MFCR2_STOP_DETECTED_WITH_SIGNAL10   0x003a
#define MFCR2_STOP_DETECTED_WITH_SIGNAL11   0x003b
#define MFCR2_STOP_DETECTED_WITH_SIGNAL12   0x003c
#define MFCR2_STOP_DETECTED_WITH_SIGNAL13   0x003d
#define MFCR2_STOP_DETECTED_WITH_SIGNAL14   0x003e
#define MFCR2_STOP_DETECTED_WITH_SIGNAL15   0x003f

/*****************************************
    MFCR2_BUSY is returned when the processing
    is still going on. Otherwise returned value
    would indicate the tone detected or error
*****************************************/

#define MFCR2_BUSY  0x0000

/*****************************************
     Structure for MFCR2 Detect Handle
*****************************************/

typedef struct
{
```


For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
    Int16 *In_Context_buf;
    UInt16 context_buf_length;
    Word16 *MFCR2_DATAStruct;
    Frac16 *Circular_Buffer;
    Frac16 *Copy_Filter_Coefs; /* Should be allocated in internal memory. */
}MFCR2_sHandle;

/*****************************************************
     Structure for MFCR2 Detect Configuration
     to be used by the user. Use the #defines
     MFCR2_FORWARD or MFCR2_BACKWARD defined in
     this file
*****************************************************/

typedef struct
{
    Int16 DIRECTION;

}MFCR2_sConfigure;

/*******************************
 MFCR2 Detect Function Prototypes
*******************************/

EXPORT MFCR2_sHandle * MFCR2DetectCreate (MFCR2_sConfigure *pConfig);

EXPORT void MFCR2DetectInit (MFCR2_sHandle *pMFCR2Detect, MFCR2_sConfigure
*pConfig);

EXPORT Result MFCR2Detect (MFCR2_sHandle *pMFCR2Detect,
                           Int16 *pSamples,
                           UInt16 NumSamples);

EXPORT void MFCR2DetectDestroy (MFCR2_sHandle *pMFCR2Detect);


/*****************************************************
 Constants used by the detect algorithm. User should not
 change this constant value
 *****************************************************/

#define FRAME_SZ 32

#endif
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3 Specifications

The following pages describe the MFCR2 Detection library functions.

Function arguments for each routine are described as *in*, *out*, or *inout*. An *in* argument means that the parameter value is an input only to the function. An *out* argument means that the parameter value is an output only from the function. An *inout* argument means that a parameter value is an input to the function, but the same parameter is also an output from the function.

Typically, *inout* parameters are input pointer variables in which the caller passes the address of a preallocated data structure to a function. The function stores its results within that data structure. The actual value of the *inout* pointer parameter is not changed.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.1 *MFCR2DetectCreate*

Call(s):

```
MFCR2_sHandle * MFCR2DetectCreate (MFCR2_sConfigure *pConfig);
```

Required Header: "*MFCR2Detect.h", "mem.h"*

Arguments:

Table 3-1. *MFCR2DetectCreate* Arguments

pConfig	*in*	Points to the configuration data for MFCR2 Detection

Description: The *MFCR2* function creates an instance of MFCR2 Detection; 216 words are dynamically allocated per instance, and multiple instances are possible. The *pConfig* argument points to a *MFCR2_sConfigure* structure, which configures MFCR2 Detection operation. Before calling *MFCR2DetectCreate,* the user must initialize the DIRECTION parameter in the *pConfig* structure to either MFCR2_FORWARD or MFCR2_BACKWARD, depending on whether the input is going to be a forward frequency combination or a backward frequency combination. The function *MFCR2DetectCreate* itself calls the initialization function, *MFCR2DetectInit*. The library dynamically allocates memory using the *mem* library, as shown in **Code Example 3-2**.

Code Example 3-2. *mem* Library

```
#include "mem.h"
#include "MFCR2Detect.h"
MFCR2_sHandle *MFCR2DetectCreate (MFCR2_sConfigure *pConfig)
{
    MFCR2_sHandle *pMFCR2Detect;

    /* Allocate the memory for the handle structure*/

    pMFCR2Detect = (MFCR2_sHandle *) memMallocEM (
                   sizeof (MFCR2_sHandle));
    if (pMFCR2Detect == NULL) return (NULL);

    /* Allocate memory for the MFCR2datastruct */
    pMFCR2Detect->MFCR2_DATAStruct = (Word16 *) memMallocIM (
                     110 * sizeof (Word16));
    if ((pMFCR2Detect->MFCR2_DATAStruct == NULL) ||
       (memIsIM (pMFCR2Detect->MFCR2_DATAStruct) == false))
    {
        MFCR2DetectDestroy (pMFCR2Detect);
        return (NULL);
    }

    /* Allocate memory for the Copy Filter coefs in the Internal memory */
    pMFCR2Detect->Copy_Filter_Coefs = (Word16 *) memMallocIM (
                                      10 * sizeof (Word16));
    if ((pMFCR2Detect->Copy_Filter_Coefs == NULL) ||
       (memIsIM (pMFCR2Detect->Copy_Filter_Coefs) == false))
    {
```


For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
        MFCR2DetectDestroy (pMFCR2Detect);
        return (NULL);
    }

    /* Allocate memory for the Circular buffer of size 64 */
    pMFCR2Detect->Circular_Buffer =
           (Frac16 *) memMallocAlignedEM (64 * sizeof (Frac16));
    if ((pMFCR2Detect->Circular_Buffer == NULL) ||
       (memIsAligned (pMFCR2Detect->Circular_Buffer,(64 * sizeof(Frac16))) == false))
    {

       MFCR2DetectDestroy (pMFCR2Detect);
       return (NULL);
    }

    /* Allocate memory for the In_Context_buf */

    pMFCR2Detect->In_Context_buf = (Int16 *) memMallocEM (FRAME_SZ * sizeof(Int16));
    if (pMFCR2Detect->In_Context_buf == NULL)
    {
        MFCR2DetectDestroy (pMFCR2Detect);
        return (NULL);
    }

    MFCR2DetectInit (pMFCR2Detect, pConfig);

    return (pMFCR2Detect);
}
```

For details on the *MFCR2_sHandle* structure, please refer to **Code Example 3-1**.

If the *MFCR2DetectCreate* function is called to create an instance, then *MFCR2DetectDestroy,* discussed in **Section 3.3.4**, should be used to destroy the instance.

Alternatively, the user can allocate memory statically, which requires duplicating all statements in the *MFCR2DetectCreate* function. In this case, the user can call the *MFCR2DetectInit* function directly, bypassing the *MFCR2DetectCreate* function. If the user dynamically allocates memory without calling *MFCR2DetectCreate,* then the user himself must destroy the memory allocated.

Returns: A pointer to an instance of MFCR2 Detection.

Special Considerations:

- If *MFCR2DetectCreate* is called, then the user need not call the *MFCR2DetectInit* function, as it is called internally in the *MFCR2DetectCreate* function.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Code Example 3-3. Use of *MFCR2DetectCreate* Interface

```
#include "MFCR2Detect.h"

void testMFCR2DetectCreate (void)
{
       MFCR2_sHandle *pMFCR2Detect;
       MFCR2_sConfigure pConfig;


       /* Initialize the pConfig structure */

       pConfig.DIRECTION = MFCR2_BACKWARD;

       pMFCR2Detect = MFCR2DetectCreate(&pConfig);
       if (pMFCR2Detect == NULL) assert(!"Cannot allocate memory");

       ....
}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.2 *MFCR2DetectInit*

Call(s):

```
void MFCR2DetectInit (MFCR2_sHandle *pMFCR2Detect,
                      MFCR2_sConfigure *pConfig)
```

Required Header: "*MFCR2Detect.h"*

Arguments:

Table 3-2. *MFCR2DetectInit* Arguments

pMFCR2Detect	*in*	Handle to an instance of MFCR2 Detection
pConfig	*in*	A pointer to a data structure containing data for initializing the MFCR2 Detection algorithm

Description: The *MFCR2DetectInit* function initializes the MFCR2 Detection algorithm. During initialization, all resources will be set to their initial values in preparation for MFCR2 Detection operation.

The parameter *pConfig* points to a data structure of type *MFCR2_sConfigure*; the *MFCR2DetectInit* function initializes the filter states, filter coefficients and other parameters used for MFCR2 detection.

Returns: None

Special Considerations:

- If *MFCR2DetectCreate* is called, then the user need not call the *MFCR2DetectInit* function, which is called internally in the *MFCR2DetectCreate* function.

Code Example 3-4. Use of *MFCR2DetectInit* Interface

```
#include "MFCR2Detect.h"

void testMFCR2det (void)
{
      /* Allocate the memory for the handle structure*/
      MFCR2_sHandle *pMFCR2Detect;

      MFCR2_sConfigure pConfig;

      /* Allocate memory for the MFCR2datastruct */
      pMFCR2Detect->MFCR2_DATAStruct = /* 110 words of internal memory */

      /* Allocate memory for the Copy Filter coefs in the Internal memory */
      pMFCR2Detect->Copy_Filter_Coefs = /* 10 words of internal memory */

      /* Allocate memory for the Circular buffer of size 64 */
      pMFCR2Detect->Circular_Buffer = /* Aligned Memory Buffer of 64 words */

      /* Allocate memory for the In_Context_buf */
      pMFCR2Detect->In_Context_buf = /* FRAME_SZ words of memory */

      /* Initialize the pConfig structure */
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
        pConfig.DIRECTION = MFCR2_BACKWARD;

        MFCR2DetectInit (pMFCR2Detect, pConfig);

        ....
}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.3 *MFCR2Detect*

Call(s):

```
Result MFCR2Detect (MFCR2_sHandle *pMFCR2Detect, Int16 *pSamples,UInt16
                    NumSamples)
```

Required Header: "*MFCR2Detect.h"*

Arguments:

Table 3-3. ,*MFCR2Detect* Arguments

pMFCR2Detect	*in*	Handle to an instance of MFCR2 Detection
pSamples	*in*	Pointer to input buffer containing samples, sampled at 8KHz. Each sample is represented in Q15 format.
NumSamples	*in*	The number of input samples

Description: The *MFCR2Detect* function processes the input samples in blocks of 32 samples and checks for the presence of a valid MFCR2 tone. **Code Example 3-5** shows the use of *MFCR2Detect* interface. Every time the *MFCR2Detect* function is called, the user must check for the return value. When the signal is yet to be detected, the function returns MFCR2_BUSY. Once the signal is detected, the function returns the number corresponding to the detected signal. For details on the exact values, please refer to **Returns,** which follows. After the signal is detected, the *MFCR2Detect* function watches for the signal to stop. Until the signal stops, the *MFCR2Detect* function keeps returning the value corresponding to the detected signal. Once the signal stops, the *MFCR2Detect* function returns the value corresponding to the signal stop. For details on the exact values, please refer to **Returns**, which follows. When there is no valid MFCR2 tone, this function returns "MFCR2_INVALID" and the calling routine may decide to stop calling *MFCR2Detect*.

Returns: The return values specify the following conditions:

MFCR2_BUSY	0x0000	Value returned when processing is occurring
MFCR2_INVALID	0x8000	This value is returned when there is no valid MFCR2 tone
MFCR2_SIGNAL1_DETECTED	0x0001	Value returned when Signal 1 is detected
MFCR2_SIGNAL2_DETECTED	0x0002	Value returned when Signal 2 is detected
MFCR2_SIGNAL3_DETECTED	0x0003	Value returned when Signal 3 is detected
MFCR2_SIGNAL4_DETECTED	0x0004	Value returned when Signal 4 is detected
MFCR2_SIGNAL5_DETECTED	0x0005	Value returned when Signal 5 is detected

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

MFCR2_SIGNAL6_DETECTED	0x0006	Value returned when Signal 6 is detected
MFCR2_SIGNAL7_DETECTED	0x0007	Value returned when Signal 7 is detected
MFCR2_SIGNAL8_DETECTED	0x0008	Value returned when Signal 8 is detected
MFCR2_SIGNAL9_DETECTED	0x0009	Value returned when Signal 9 is detected
MFCR2_SIGNAL10_DETECTED	0x000a	Value returned when Signal 10 is detected
MFCR2_SIGNAL11_DETECTED	0x000b	Value returned when Signal 11 is detected
MFCR2_SIGNAL12_DETECTED	0x000c	Value returned when Signal 12 is detected
MFCR2_SIGNAL13_DETECTED	0x000d	Value returned when Signal 13 is detected
MFCR2_SIGNAL14_DETECTED	0x000e	Value returned when Signal 14 is detected
MFCR2_SIGNAL15_DETECTED	0x000f	Value returned when Signal 15 is detected
MFCR2_STOP_DETECTED_WITH_SIGNAL1	0x0031	Value returned when Signal 1 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL2	0x0032	Value returned when Signal 2stops
MFCR2_STOP_DETECTED_WITH_SIGNAL3	0x0033	Value returned when Signal 3stops
MFCR2_STOP_DETECTED_WITH_SIGNAL4	0x0034	Value returned when Signal 4stops
MFCR2_STOP_DETECTED_WITH_SIGNAL5	0x0035	Value returned when Signal 5stops
MFCR2_STOP_DETECTED_WITH_SIGNAL6	0x0036	Value returned when Signal 6 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL7	0x0037	Value returned when Signal 7 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL8	0x0038	Value returned when Signal 8 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL9	0x0039	Value returned when Signal 9 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL10	0x003a	Value returned when Signal 10 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL11	0x003b	Value returned when Signal 11 stops

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

MFCR2_STOP_DETECTED_WITH_SIGNAL12	0x003c	Value returned when Signal 12 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL13	0x003d	Value returned when Signal 13 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL14	0x003e	Value returned when Signal 14 stops
MFCR2_STOP_DETECTED_WITH_SIGNAL15	0x003f	Value returned when Signal 15 stops

MFCR2_BUSY is returned when the processing is still continuing. Otherwise, the returned value would indicate either the tone detected or an error

Special Considerations: The *MFCR2Detect* function expects input as 16-bit linear PCM samples, sampled at 8KHz only.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Code Example 3-5. Use of *MFCR2Detect* Interface

```
#include "MFCR2Detect.h"
#include <stdio.h>

void testMFCR2det (void)
{
       MFCR2_sHandle *pMFCR2Detect;
       MFCR2_sConfigure pConfig;
       Int16 *SampleBuf;
       UInt16 NumSamples;
       Result result;
       Int16 Loop=1;

       /* Initialize the pConfig structure */

       pConfig.DIRECTION = BACKWARD;

       pMFCR2Detect = MFCR2DetectCreate(&pConfig);
       if (pMFCR2Detect[i] == NULL) assert(!"Cannot allocate memory");

       ....
       ....

       NumSamples = 32;
       result = 0;
       while(Loop)
       {
              /* Fill the input buffer. In this test, we fill the input buffer,
              SampleBuf, with samples corresponding to Signal 1 */

              result = MFCR2Detect(pMFCR2Detect, SampleBuf, NumSamples);
              if (result == MFCR2_BUSY) continue;
              else if (result == MFCR2_SIGNAL1_DETECTED)
              {
```


For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
                        printf("Signal detected correctly");
                        Loop = 0;
                }
                else
                {
                        printf("Test Failed... Signal not detected correctly");
                        exit(0);
                }
        }
        Loop = 1;
        while(Loop)
        {
                /* Fill the input buffer*/
                result = MFCR2Detect(pMFCR2Detect, SampleBuf, NumSamples);
                if (result == MFCR2_SIGNAL1_DETECTED) continue;
                else if (result == MFCR2_STOP_DETECTED_WITH_SIGNAL1)
                {
                        printf("Signal stop detected correctly");
                        Loop = 0;
                }
                else
                {
                        printf("Test Failed... Signal stop not detected correctly");
                        exit(0);
                }
        }
}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.4 *MFCR2DetectDestroy*

Call(s):

```
void MFCR2DetectDestroy (MFCR2_sHandle *pMFCR2Detect)
```

Required Header: "*MFCR2Detect.h"* and "*mem.h"*

Arguments:

Table 3-4. *MFCR2DetectDestroy* Arguments

pMFCR2Detect	*in*	Handle to an instance of MFCR2 Detection generated by a call to *MFCR2DetectCreate*

Description: The *MFCR2DetectDestroy* function frees the instance of MFCR2 Detection originally created by a call to *MFCR2DetectCreate.*

Returns: None

Special Considerations: None

Code Example 3-6. Use of *MFCR2DetectDestroy* Interface

```
#include "MFCR2Detect.h"
#include "mem.h"

void testMFCR2det (void)
{
      MFCR2_sHandle *pMFCR2Detect;
      MFCR2_sConfigure pConfig;
      Int16 *SampleBuf;
      UInt16 NumSamples;
      Result result;

      /* Initialize the pConfig structure */

      pConfig.DIRECTION = BACKWARD;

      pMFCR2Detect = MFCR2DetectCreate(&pConfig);
      if (pMFCR2Detect[i] == NULL) assert(!"Cannot allocate memory");

      ....

      /* Fill the input buffer*/
      ....

      NumSamples = 32;
      result = 0;

      result = MFCR2Detect(pMFCR2Detect, SampleBuf, NumSamples);
      ....
      MFCR2DetectDestroy(pMFCR2Detect);
}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Chapter 4
Building the MFCR2 Detection Library

4.1 Building the MFCR2 Detection Library

The MFCR2 Detection library combines all of the components described in previous sections into one library: *mfcr2.lib*. To build this library, a Metrowerks' CodeWarrior project, *MFCR2.mcp*, is provided. This project and all components necessary to build the MFCR2 Detection library are located in the *....\nos\telephony\MFCR2* directory of the SDK directory structure.

There are two methods to execute the system library project build: dependency build and direct build.

4.1.1 Dependency Build

Dependency build is the easiest approach and requires no additional work on the user's part. If you add the MFCR2 Detection library project, *MFCR2.mcp*, to your application project, as shown in **Figure 4-1**, the MFCR2 Detection library will automatically build when the application is built.

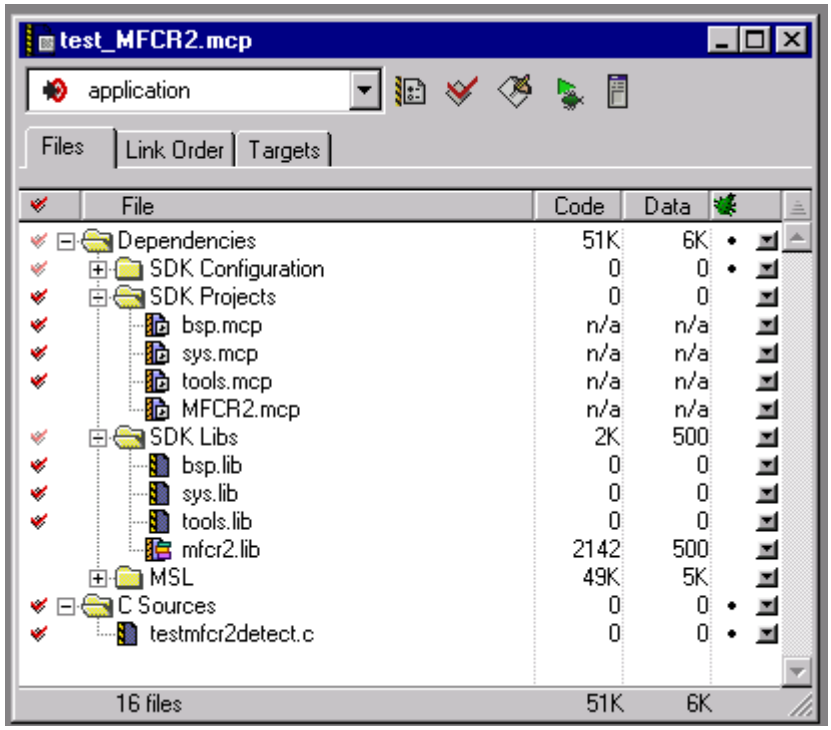

Figure 4-1. Dependency Build for MFCR2 Detection Library

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

4.1.2 Direct Build

Direct build allows you to build the MFCR2 Detection library independently of any other build. Follow these steps:

Step 1. Open *MFCR2.mcp* project, as shown in **Figure 4-2**.

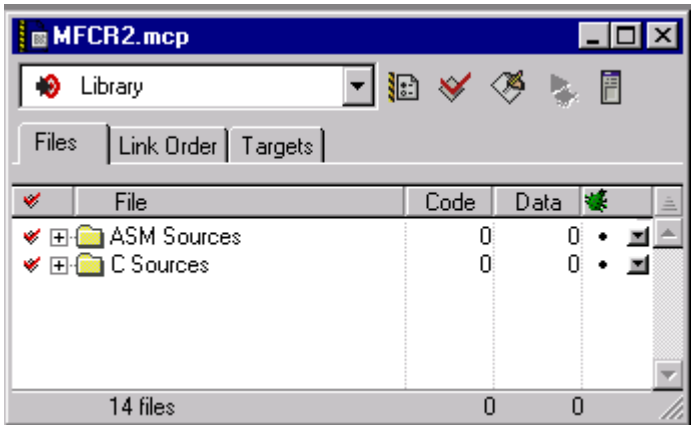

Figure 4-2. MFCR2 Detection Project

Step 2. Execute the build by pressing function key [*F7*] or by choosing the *Make* command from the *Project* menu; see **Figure 4-3**.

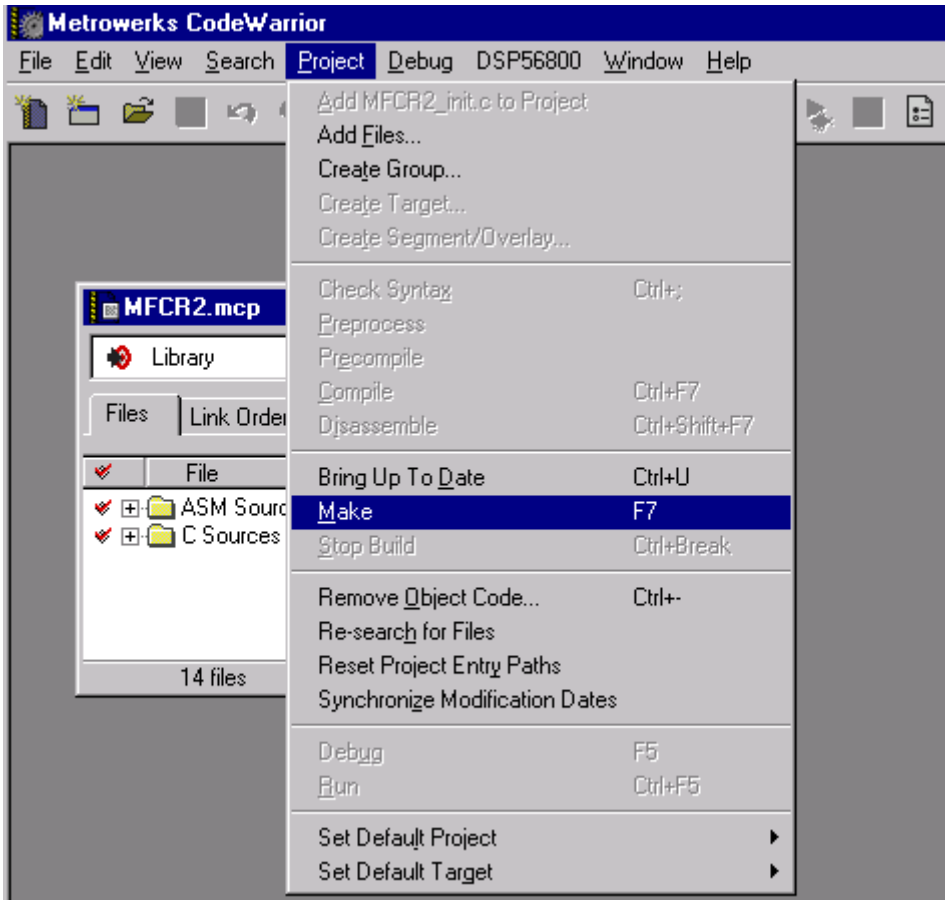

Figure 4-3. Execute Make

For More Information On This Product,
Go to: www.freescale.com
Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

At this point, if the build is successful, an *mfcr2.lib* library file is created in the directory *...\nos\telephony\MFCR2\Debug*.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 5
Linking Applications with the MFCR2 Detection Library

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

5.1 MFCR2 Detection Library

The MFCR2 Detection library consists of APIs, which provide an interface between the user application and the MFCR2 Detection modules. To invoke MFCR2 Detection, the following APIs must be called in this order:

- *MFCR2DetectCreate* (.....); /* Should be called only once */
- *MFCR2DetectInit* (.....); /* Called within the *MFCR2Detectcreate* function */
- *MFCR2Detect* (.....); /* Can be called as many times as the number of samples detected*/
- *MFCR2DetectDestroy* (.....); /* Should be called only once */

5.1.1 Library Sections

See **Code Example 5-1** for a sample *linker.cmd* file, which may be used for testing the MFCR2 Detection library.

Code Example 5-1. *linker.cmd* File

```
Linker.cmd file for DSP56824EVM External RAM
#        using both internal and external data memory (EX = 0)
#    and using external program memory (Mode = 3)

MEMORY {

       .pvec(RWX) : ORIGIN = 0x0000, LENGTH = 0x002C
                    # interrupt vector table ( 22 * 2 )
       .pram(RWX) : ORIGIN = 0x002C, LENGTH = 0xFFD4
                    # external program memory
       .avail(RW) : ORIGIN = 0x0000, LENGTH = 0x0030  # available
       .cwregs(RW) : ORIGIN = 0x0030, LENGTH = 0x0010
                    # C temp registrs in CodeWarrior
       .im1(RW)  : ORIGIN = 0x0040, LENGTH = 0x07C0  # data 1
       .rom(R)   : ORIGIN = 0x0800, LENGTH = 0x0800  # internal data ROM
```


For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
        .im2(RW)  : ORIGIN = 0x1000, LENGTH = 0x0600  # data 2
        .hole(R)   : ORIGIN = 0x1600, LENGTH = 0x0A00  # hole
        .data(RW)  : ORIGIN = 0x2000, LENGTH = 0xC000  # data segment
        .em(RW)  : ORIGIN = 0xE000, LENGTH = 0x1000  # data 3
        .stack(RW)  : ORIGIN = 0xF000, LENGTH = 0x0F80  # stack
        .onchip1(RW) : ORIGIN = 0xFF80, LENGTH = 0x0040
        .onchip2(RW)  : ORIGIN = 0xFFC0, LENGTH = 0x0040
}

FORCE_ACTIVE {FconfigInterruptVector}

SECTIONS {
        # Data (X) Memory Layout
        _EX_BIT     = 0;

        # Internal Memory Partitions (for mem.h partitions)
        _NUM_IM_PARTITIONS = 1;  # .im1 and .im2

        # External Memory Partition (for mem.h partitions)
        _NUM_EM_PARTITIONS = 1;   # .em

        .main_application_vector :
        {
        # .text sections

        #  vector.c MUST be placed into .pvec, otherwise the Interrupt Vector
        #  configInterruptVector will not be located at the correct address, P:0x0000

        vector.c (.text)

        > .pvec

        .main_application_code :
        {
              # .text sections

        * (.text)
        * (rtlib.text)
        * (fp_engine.text)
        * (user.text)
        } > .pram

        .main_application_data :
        {
        #
        # Define variables for C initialization code
        #
        F_Xdata_start_addr_in_ROM = ADDR(.rom) + SIZEOF(.rom) / 2;
        F_StackAddr               = ADDR(.stack);
        F_StackEndAddr            = ADDR(.stack) + SIZEOF(.stack) / 2  - 1;

        F_Xdata_start_addr_in_RAM = .;

        #
        # Memory layout data for SDK INCLUDE_MEMORY (mem.h) support
        #
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
FmemEXbit = .;
WRITEH(_EX_BIT);
FmemNumIMpartitions = .;
WRITEH(_NUM_IM_PARTITIONS);
FmemNumEMpartitions = .;
WRITEH(_NUM_EM_PARTITIONS);
FmemIMpartitionList = .;
#WRITEH(ADDR(.im1));
#WRITEH(SIZEOF(.im1) / 2);
WRITEH(ADDR(.im2));
WRITEH(SIZEOF(.im2) / 2);
FmemEMpartitionList = .;
WRITEH(ADDR(.em));
WRITEH(SIZEOF(.em) /2);

# .data sections

* (.data)
* (fp_state.data)
* (rtlib.data)

F_Xdata_ROMtoRAM_length = 0;

F_bss_start_addr =.;
_BSS_ADDR = .;

* (rtlib.bss.lo)
* (.bss)

F_bss_length = . - _BSS_ADDR;  # Copy DATA

> .data

FArchIO   = ADDR(.onchip2);

# MFCR2 detect internal data starts here
#--------------------------------------

.MFCR2_detect_int_data :
{

* ( MFCR2_DETECT_INT_MEM.data)
* ( MFCR2_DETECT_INT_MEM.bss)

} > .im1

# MFCR2 detect internal data ends here
#--------------------------------------
}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 6
MFCR2 Detection Applications

6.1 Test and Demo Applications

To verify the MFCR2 Detection algorithm, a test application has been developed. Refer to the **Targeting Motorola DSP568xx Platform** Manual for the DSP you are using to see if the test application is available for your target.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 7
License

7.1 Limited Use License Agreement

LIMITED USE LICENSE AGREEMENT

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THIS SOFTWARE. BY USING OR COPYING THE SOFTWARE, YOU AGREE TO THE TERMS OF THIS AGREEMENT.

The software in either source code form ("Source") or object code form ("Object") (cumulatively hereinafter "Software") is provided under a license agreement ("Agreement") as described herein. Any use of the Software including copying, modifying, or installing the Software so that it is usable by or accessible by a central processing unit constitutes acceptance of the terms of the Agreement by the person or persons making such use or, if employed, the employer thereof ("Licensee") and if employed, the person(s) making such use hereby warrants that they have the authority of their employer to enter this license agreement,. If Licensee does not agree with and accept the terms of this Agreement, Licensee must return or destroy any media containing the Software or materials related thereto, and destroy all copies of the Software.

The Software is licensed to Licensee by Motorola Incorporated ("Motorola") for use under the terms of this Agreement. Motorola retains ownership of the Software. Motorola grants only the rights specifically granted in this Agreement and grants no other rights. Title to the Software, all copies thereof and all rights therein, including all rights in any intellectual property including patents, copyrights, and trade secrets applicable thereto, shall remain vested in Motorola.

For the Source, Motorola grants Licensee a personal, non-exclusive, non-assignable, revocable, royalty-free right to use, copy, and make derivatives of the Source solely in a development system environment in order to produce object code solely for operating on a Motorola semiconductor device having a central processing unit ("Derivative Object").

For the Object and Derivative Object, Motorola grants Licensee a personal, non-exclusive, non-assignable, revocable, royalty-free right to copy, use, and distribute the Object and the Derivative Object solely for operating on a Motorola semiconductor device having a central processing unit.

Licensee agrees to: (a) not use, modify, or copy the Software except as expressly provided herein, (b) not distribute, disclose, transfer, sell, assign, rent, lease, or otherwise make available the Software, any derivatives thereof, or this license to a third party except as expressly provided herein, (c) not remove obliterate, or otherwise defeat any copyright, trademark, patent or proprietary notices, related to the

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Software (d) not in any form export, re-export, resell, ship or divert or cause to be exported, re-exported, resold, shipped, or diverted, directly or indirectly, the Software or a direct product thereof to any country which the United States government or any agency thereof at the time of export or re-export requires an export license or other government approval without first obtaining such license or approval.

THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS AND WITHOUT WARRANTY OF ANY KIND INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MOTOROLA BE LIABLE FOR ANY LIABILITY OR DAMAGES OF ANY KIND INCLUDING, WITHOUT LIMITATION, DIRECT OR INDIRECT OR INCIDENTAL OR CONSEQUENTIAL OR PUNITIVE DAMAGES OR LOST PROFITS OR LOSS OF USE ARISING FROM USE OF THE SOFTWARE OR THE PRODUCT REGARDLESS OF THE FORM OF ACTION OR THEORY OF LIABILITY (INCLUDING WITHOUT LIMITATION, ACTION IN CONTRACT, NEGLIGENCE, OR PRODUCT LIABILITY) EVEN IF MOTOROLA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THIS DISCLAIMER OF WARRANTY EXTENDS TO LICENSEE OR USERS OF PRODUCTS AND IS IN LIEU OF ALL WARRANTIES WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE.

Motorola does not represent or warrant that the Software is free of infringement of any third party patents, copyrights, trade secrets, or other intellectual property rights or that Motorola has the right to grant the licenses contained herein. Motorola does not represent or warrant that the Software is free of defect, or that it meets any particular requirements or need of the Licensee, or that it conforms to any documentation, or that it meets any standards.

Motorola shall not be responsible to maintain the Software, provide upgrades to the Software, or provide any field service of the Software. Motorola reserves the right to make changes to the Software without further notice to Licensee.

The Software is not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Software could create a situation where personal injury or death may occur. Should Licensee purchase or use the Software for any such unintended or unauthorized application, Licensee shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the Software.

The term of this Agreement is for as long as Licensee uses the Software for its intended purpose and is not in default of any provisions of this Agreement. Motorola may terminate this Agreement if Licensee is in default of any of the terms and conditions of this Agreement.

This Agreement shall be governed by and construed in accordance with the laws of the State of Arizona and can only be modified in a writing signed by both parties. Licensee agrees to jurisdiction and venue in the State of Arizona.

By using, modifying, installing, compiling, or copying the Software, Licensee acknowledges that this Agreement has been read and understood and agrees to be bound by its terms and conditions. Licensee agrees that this Agreement is the complete and exclusive statement of the agreement between Licensee and Motorola and supersedes any earlier proposal or prior arrangement, whether oral or written, and any other communications relative to the subject matter of this Agreement.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Index

D

Digital Signal Processor
    DSP xi
DSP xi

I

I/O xi
IDE xi
Input/Output
    I/O xi
Integrated Development Environment
    IDE xi

M

Metrowerks CodeWarrior 4-1
MFCR2 xi
MFCR2 Detection library project
    MFCR2.mcp 4-1
Mfcr2 directory 2-3
MFCR2.mcp 4-1
MFCR2DetectDestroy 3-14
Million Instructions Per Second
    MIPS xi
MIPS xi
Motorola DSP56800 Family Manual xi
Motorola DSP56800E Reference Manual xi
Motorola DSP568xx User's Manual xi
Motorola Embedded SDK Programmer's Guide xi
Multi-Frequency Compelled Register Signaling
    MFCR2 xi

O

OnCE xi
On-Chip Emulation
    OnCE xi

S

SDK xi
Sotfware Development Kit
    SDK xi
Source
    SRC xi
SRC xi

T

Targeting Motorola DSP568xx User's Manual xi
telephony 2-2

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and the Stylized M Logo are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

How to reach us:
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1–303–675–2140 or 1–800–441–2447

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3–20–1, Minami–Azabu. Minato–ku, Tokyo 106–8573 Japan. 81–3–3440–3569

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852–26668334

Technical Information Center: 1–800–521–6274

HOME PAGE: http://www.motorola.com/semiconductors/

MOTOROLA

For More Information On This Product,
Go to: www.freescale.com

SDK151/D